The Eighteenth International Symposium on Artificial Life and Robotics 2013 (AROB 18th ’13),
Daejeon Convention Center, Daejeon, Korea, January 30-February 1, 2013

# Object Tracking Based on Adaboost Classifier and Particle Filter

Chin-Lun Lai[1] , Li-Yin Lee[2]

[1,2]Communication Engineering Department, Oriental Institute of Technology, Taipei, Taiwan

[1]fo001@mail.oit.edu.tw
[2]990130108@mail.oit.edu.tw

**Abstract:** In this paper, a hybrid structure combining adaboost classifier and particle filter is proposed to automatically detect and track the pedestrian targets. The adaboost detection process is adopted first to target candidate objects, and then the particle filter is applied for confirming and tracking of targets. Experiment results show that via the proposed method, the drawback of the current particle filters which requires specifying an object to be tracked in advance can be overcome, while performing good also in cases of target missing, occlusion, and identifying the previously appeared objects.

**Keywords**：object tracking, adaboost detection, particle filter

## 1 INTRODUCTION

Application of object tracking has always been an important issue in computer vision or image processing applications. In the early stages, object tracking had been applied to air traffic control. Recently, it has often been applied to with security monitoring related fields. There are various types of methods for object tracking. Generally, these methods can be divided into time domain methods and space domain methods. Time domain methods include such as motion estimation[1] or optical flow, etc., while space domain methods include k-means, mean-shift, kalman filter, and particle filters [2].

In a time domain system, the target must be able to show time differences. In other words, the target has to move so that judgments can be made. In a space domain system, judgments are made based on image characteristics of the targets. And usually judging methods based on characteristic information are more complex and diversified.

However, if only a time domain method is applied, the only thing that can be confirmed is that whether the target is moving or not. It is difficult to find out if this target is the interest one. Methods like k-means and mean-shift require manual settings and object size changes are not allowed. Kalman filter is used for linear systems thus is not appropriate for real-world systems which are usually no-linear systems. On the other hand, particle filter is a good solution for target tracking but is only applicable when the scene is known and possible locations of the target are preset[2]. However, adding the adaboost algorithm helps to solve this issue [3]. Therefore, this study proposed a combined structure with adaboost detection and particle filtering method to resolve the problems mentioned above for the pedestrian tracking problems.

## 2 SYSTEM FRAMEWORK

The proposed system framework is shown in figure 1 and the detailed steps are described as follows:

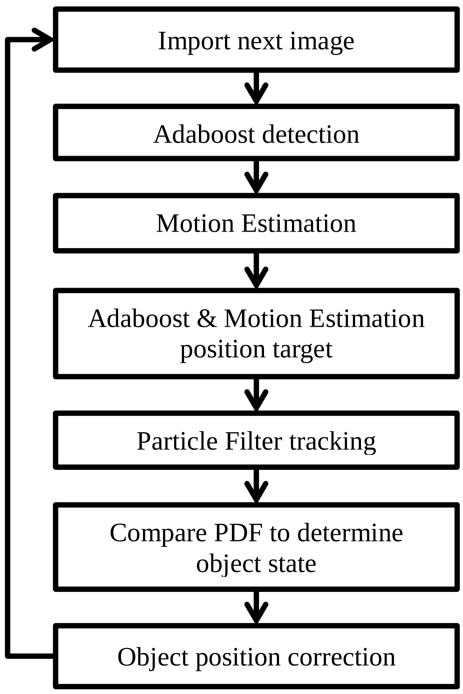



**Fig. 1.** system framework

### 2.1 Adaboost detection

To improve the prediction rate and reduce the error rate, four adaboost cascades were combined to form the initial conditions for pedestrian detection. The types of the four cascades were upper body, lower body, front side of whole body, and back side of whole body. All four cascades were objects or parts of objects, resolving the issue of imprecise detection with only one cascade.

The training sequences included 2094 positive samples and 1333 negative samples. The width and height were 10 and 24 (pels), using all upright rectangle feature and 45 rotated rectangle feature [4].

Two of the four cascades were used to detect upper body and lower body. Therefore, whole body normalization was required. An upper body includes head and shoulders. Thus, a bounding box may contain head and shoulders. And in the cascade training, the size was 22x20, with the shape being close to a square. Since humans' upper bodies are rectangles, it is necessary to normalize bounding boxes for upper bodies as follows:

$$\mathrm{BBox}_{\mathrm{up(x)}} = \mathrm{upbody(x)} + \frac{\mathrm{upbody(w)}}{4} \tag{1a}$$

$$\mathrm{BBox}_{\mathrm{up(w)}} = \frac{3}{4} \times \mathrm{upbody(w)} \tag{1b}$$

$$\mathrm{BBox}_{\mathrm{up(h)}} = \sqrt{2} \times \mathrm{upbody(h)} \tag{1c}$$

where $BBox_{up}$ is the normalized bounding box and *upbody* is the bounding box detected using adaboost. Parameters for upbody, (x, y, w, h), represent the x-coordinate, y-coordinate, width, and height, respectively.

Similarly, as for the lower body training, the size was 19x23. Thus whole body normalization was performed by (2) as follows.

$$\mathrm{BBox}_{\mathrm{low(y)}} = \mathrm{lowbody(y)} - \frac{1}{2}\mathrm{lowbody(h)} \tag{2a}$$

$$\mathrm{BBox}_{\mathrm{down(h)}} = \sqrt{2} \times \mathrm{lowbody(h)} \tag{2b}$$

where $BBox_{low}$ is the normalized bounding box and *lowbody* is the bounding box detected using adaboost.

### 2.2 Motion estimation

By adopting adaboost, many candidate targets were found hence further filtering was required to remove the misjudged ones. It can be done by adding simple motion estimation process. First, binary threshold for three continuous images was calculated. And then morphology erosion and closing, with 3x3 and 5x5 masks, were applied to these 3 images to obtain a binary image using OR gate, in order to determine the moving range of an object.

### 2.3 Adaboost & motion estimation

After applying motion estimation to the normalized bounding boxes, the areas of the bounding boxes are calculated. The bounding boxes of small sizes were considered as noises, while the centers of those which were not noises were obtained.

With a given size threshold, the bounding boxes with areas over the average were removed. Among the bounding boxes left, the one with the smallest area was considered as the most possible target. Then, target positioning was performed with the most possible bounding box. And the positioning information, including the center and the range of the target, was sent to the particle filter.

### 2.4 Particle filter tracking

Ones the object was confirmed, the particles of the particle filter are initialized using the positioning information of the object to perform particle filter tracking[2]. The re-sampling method adopted was the systematic method[5]. The new model was based on the average of the two previous models.

After that, the PDF of the target at *t-1* was compared with that at *t* through the corresponding correlation coefficient. If the difference was too large, the target was considered disappeared or lost. In that case, adaboost was applied again for re-positioning.

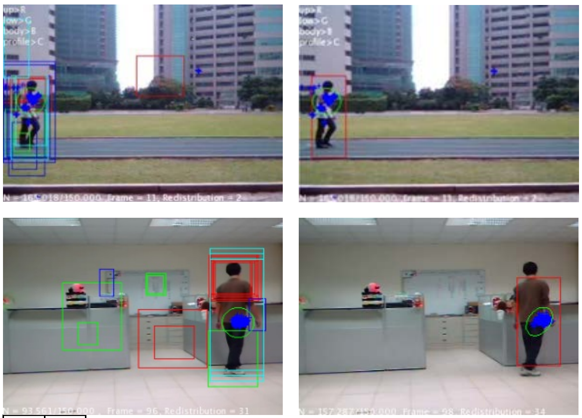
### 2.5 Adaboost & particle filter correction

In order to prevent particle distribution from being too scattered due to large changes of PDF of the target, adjustments needed to be made based on the adaboost detection results when applying particle filter. In continuous images of the target, the functions of the size and the distance were supposed to be continuous. Therefore, by referencing continuous sizes and distances from adaboost and particle filter, targets of abnormal sizes were removed and targets of large distances were repositioned according to corresponding adaboost information while particle filter was reinitialized.

## 3 EXPERIMENTAL RESULTS AND DISCUSSIONS

Two detection results with different scenes are shown in figure 2. It is noted that, in figure 2(a)(c), squares with different colors represent the adaboost detection results corresponding to different training database, where red, green, blue, and cyan squares resulted from upper bodies, lower bodies, front/back of whole body, and sideways of whole body respectively.

In figure 2(b)(d), the red squares are generated from (a) and (c) after filtering, while the blue points and green ellipses represent the particles and average statuses of particles respectively.



| | |
|---|---|
| a | b |
| c | d |

**Fig. 2.**(a)(c) original image with adaboost detection results, and (b)(d): the corresponding final detection result and target location of particles.

Figure 3 shows the number of detection and that of false alarm for each method in scene1.

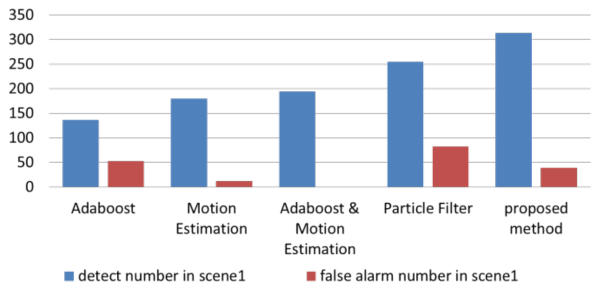



**Fig. 3.** Detection and false alarm results of compared algorithm in test scene 1.

Figure 4 shows the number of detection and that of false alarm for each method in scene2.

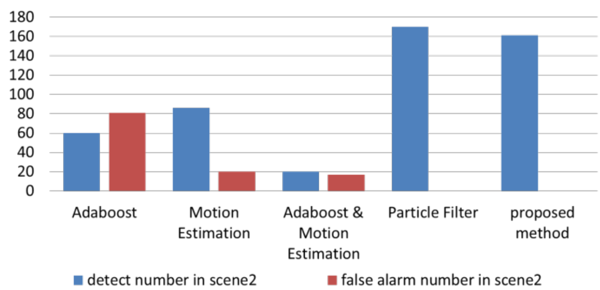


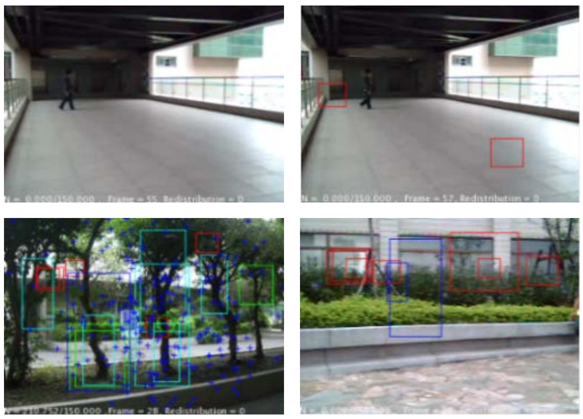

**Fig. 4.** Detection and false alarm results of compared algorithm in test scene 2.

The statistical result of the detection and false alarm rate is shown in Table 1 to illustrate the efficiency differences between the proposed method and the current tracking methods. The detection rate is defined as: the number of successful tracking of the object in the 507 images (there're a total of 701 images from the videos for the two testing scenes, and after the ones with the target disappearing were removed, a total of 507 images were left).

**Table 1.** The detection rate and false alarm rate of compared algorithms in test scenarios

| Method | Detection rate | False alarm rate |
|---|---|---|
| Adaboost[3] | 0.4065 | 0.1572 |
| Motion Estimation[1] | 0.5341 | 0 |
| Adaboost & Motion Estimation | 0.5756 | 0.0356 |
| particle filter[2] | 0.7566 | 0.2433 |
| proposed method | 0.9139 | 0.1483 |

According to Table 1, compared with other algorithms, the detection rate of the proposed method was higher and the false alarm rate was lower. However, its performances in other experiment scenes weren't all as outstanding. For example, in a scene with complex environment and weak lights, as shown in figure 5, tracking was not possible. The reason was the high error rate and low accuracy of adaboost, making successful positioning impossible. In the future, before adaboost is applied, preprocessing using histogram equalization can be done to reduce issues caused by weak lights and low contrast. In addition, by increasing the number of negative sample for cascade training, more features which don't belong to the object can be learned to break the limitation of this system.



| | |
|---|---|
| a | b |
| c | d |

**Fig. 5.** Failure positioning examples. (a)(b) low contrast conditions, and (c)(d) complex environment. In both cases the adaboost results low detection rate and high false alarm.

## 4 CONCLUSIONS

An efficient and effective pedestrian tracking method is proposed to overcome the drawbacks of the existing methods based on particle filter. With adaboost appended, the positioning information can be obtained in advance for particle filter initiation while the tracking results can be adjusted automatically. Via this structure the limitation of having to specifying target positions in advance manually is broken and it also solved issues such as object disappearing, object being masked, form changes, and object re-appearing. Therefore, this method can be applied in various scenes thus can be widely applied in different areas with high efficiency.

Like other current methods, the issue of bad performances in detection and tracking in a complex environment still exists with the object tracking method proposed by this study. In the future, we will add a preprocessing step of image enhancement before target detection and increase the number of negative samples in the sample for cascade training, to solve the issue above, so this system can be applied more widely with efficiency.

## REFERENCES

[1] Saxena V., Grover S., and Joshi S.(2008), A real time face tracking system using rank deficient face detection and motion estimation. Cybernetic Intelligent Systems, 2008, pp.1-6

[2] Katja Nummiaro, Esther Koller-Meier, and Luc Van Gool.(2002), An Adaptive Color-Based Particle Filter", Image and Vision Computing, 2002, pp.1-22

[3] Viola, P., and Jones M.(2001), Robust Real-Time Face Detection. Computer Vision, ICCV 2001. vol 2, 2001, pp. 137-154

[4] Lienhart R., and Maydt J.(2002) An Extended Set of Haar-like Features for Rapid Object Detection", Image Processing, vol. 1, 2002, pp.I-900 - I-903

[5] Gordon N, Salmo D J, and Smith A F M(1993), Novel approach to nonlinear/ non-Gaussian Bayesian state estimation, IEE Proceedings F Radar and Signal Processing, 1993, pp.107-113